REMARKS

Claims 1-19 are currently pending in the subject application and are presently under consideration. Newly added claim 38 has been appended to the listing of the claims as shown on pp. 5 of the Reply.

The Examiner is thanked for courtesies extended during an interview conducted on January 14, 2008. The focus of the interview was on deficiencies of the 35 U.S.C. §101 and 35 U.S.C. §103 rejections. While the presented matter generally related to all the claims, the crux was upon independent claims 1 and 19 and dependent claim 2. In particular, the Kuznetsov, Omoigui *et al.*, and ADO.NET references were discussed in the interview. The examiner suggested that recitation of hardware in the independent claims (*e.g.*, memory, processor or the like) could be sufficient to overcome the 35 U.S.C. §101 rejection. The examiner ceded that, in regard to the 35 U.S.C. §103 rejections, application of the Omoigui *et al.* reference to the subject matter recited in claim 2 could possibly be overcome by sufficiently pointing out distinctions between the cited portions of Omoigui *et al.* with respect to the elements of claim 2.

In response to the discussions, claims 1 and 19 are presently amended to recite hardware (*e.g.*, memory) to overcome the 35 U.S.C. §101 rejection. No agreement was reached with regard to the Kuznetsov reference or the ADO.NET reference. The interview was conducted with Ronald Krosky (Reg. No. 58,564), Matthew Clapper (technical assistant), and Examiner Hillery.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-19 Under 35 U.S.C. §101

Claims 1-19 stand rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter.

Withdrawal of this rejection is requested for at least the following reason. Claims 1-19 produce a useful, concrete, and tangible result.

> Because the claimed process applies the Boolean principle [abstract idea] ***to produce a useful, concrete, tangible result*** … on its face the claimed process comfortably falls within the scope of §101. *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed. Cir. 1999) (Emphasis added); *See State Street Bank &*

> *Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1601 (Fed.Cir.1998). The inquiry into patentability requires an examination of the contested claims to see if the claimed subject matter, as a whole, is a disembodied mathematical concept representing nothing more than a "law of nature" or an "abstract idea," or if the mathematical concept has been ***reduced to some practical application rendering it "useful***." *AT&T* at 1357 *citing In re Alappat*, 33 F.3d 1526, 31 1544, 31 U.S.P.Q.2D (BNA) 1545, 1557 (Fed. Cir. 1994) (emphasis added).

Applicants' claimed invention relates to providing a streaming input and streaming output incremental XML transformer that can be employed in push and/or pull model processing. The XML transformer facilitates a user incrementally building the output from XML data so that only a subset of an XML document needs to be loaded into memory to perform a selective transformation. Independent claims 1 and 19 recite similar limitations, namely: ***a transformer*** *that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats*; *and an output manager that facilitates at least one of* ***selectively pulling and pushing a subset of the one or more input XML items, the subset of the one or more XML items is less than the whole one or more input XML items***. In addition, independent claims 1 and 19 have been amended to recite ***memory*** configured to receive and store input XML items. Moreover, newly submitted claim 38 recites at least ***a transformer*** *that receives a plurality of input XML items and transforms the input XML items from a first format to one or more second XML formats;* ***memory*** *configured to receive and store a stream of XML items* and ***an output manager*** *that facilitates at least one of* ***selectively pulling and pushing a subset of the transformed XML items****....*

In the subject Office Action, it is contended that the claims do not produce a concrete, useful and tangible result (Office Action dated October 31, 2007, p. 2, item # 5, 1st and 2nd paragraphs). More specifically, the subject Office Action contends that the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulated data. The Office Action seems to equate transforming one or more input XML items into a different XML format, and selectively pulling or pushing a part of the input XML items done, *e.g.*, as a result of implementing software code on a suitable processor, with a thought, or a computation in the abstract. The Court of Appeals for the Federal Circuit has clearly stated that

software code and related processes are patentable subject matter under 35 U.S.C. §101, however. Specifically, the holding of the Court of Appeals for the Federal Circuit in *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325 (Fed. Cir. 2005):

> Title 35, section 101, explains that an invention includes "any new and useful process, machine, manufacture or composition of matter." ... Without question, ***software code alone qualifies as an invention eligible for patenting under these categories***, at least as processes. *Id.* at 1338 (emphasis added).

Software code alone qualifies as an invention eligible for patenting, and therefore is not to be interpreted as abstract in nature. The Office Action admits that the claimed subject matter provides for transforming a selective subset of data items, specifically a transformer or transformation component *that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats*. Applicants' representative submits that all that is relevant in light of *Eolas* is the fact that software code is received, processed, and its output facilitated for the claimed invention to produce a useful, concrete, and tangible result.

In addition to the foregoing, the Court of Appeals for the Federal Circuit has also held that application of a Boolean principle, which is an abstract idea, to data to produce a useful, concrete and tangible result provides patentable subject matter. Specifically, the holding of the Court of Appeals for the Federal Circuit in *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed. Cir. 1999):

> Because the claimed process applies the Boolean principle [abstract idea] *to produce a useful, concrete, tangible result* ... on its face the claimed process comfortably falls within the scope of §101.

It is submitted that transforming XML items involves application of a function to data analogous to *At&T Corp.* For example, in this instance the function can be a result of actions provided by an XSL style sheet implemented by a processor to transform XML data in a first format to transformed XML data in one or more second XML formats, as recited by the claims in issue. It is well known that XML items involve data having markup information providing context for the data. It is submitted that all data has utility, and thus providing selective transformation of a

subset of items to reduce processing or memory requirements associated with retrieving XML text provides a useful, concrete and tangible result for the data.

The Office Action also contends that a useful result is not produced because, "the claimed subject matter relates only to transforming data items." (Office Action dated October 31, 2007, p. 3, 1st paragraph). Applicants' representative notes that the Office Action fails to acknowledge that more than transformation is claimed in respect of utility. Specifically, *an output manager that facilitates at least one of selectively pulling and pushing a subset of the one or more input XML items* is also provided. The claimed subject matter not only provides for transforming XML items but for selectively facilitating output of a subset of the transformed XML items. In light of *Eolas*, the claimed subject matter is clearly sufficient to meet the requirements for statutory subject matter under 35 U.S.C. §101.

In addition to the foregoing, independent claims 1 and 19 are presently amended in view of the interview of January 14, 2008 with Examiner Hillery. Specifically, independent claims 1 and 19, as well as new claim 38, recite a ***memory*** configured to receive and store an(one or more) input XML item(s). New claim 38 also recites similar language. Support within the written description for this amendment is contained at least at page 4 lines 27-30: "The transformer facilitates a user incrementally building the output from XML data so that only a subset of an XML document needs to be loaded into memory to perform a selective transformation." (Applicants' specification, p. 4, lines 27-30). Because memory and data stored in memory involves computer hardware, a tangible, concrete and useful result is provided by independent claims 1 and 19, as amended, and new independent claim 38.

In view of at least the foregoing, it is readily apparent that the subject claim sets forth a useful, concrete and tangible result. Accordingly, withdrawal of this rejection is requested.

II. Rejection of Claims 1, 3, 4, 5 and 19 Under 35 U.S.C. §103(a)

Claims 1, 3, 4, 5 and 19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kuznetsov (US 6,772,413 B2). This rejection should be withdrawn for at least the following reason: Kuznetsov does not disclose or suggest each and every limitation set forth in the subject claims.

> If a reference is cited that requires some modification in order to meet the claimed invention or requires some modification in order to be properly combined with another reference and such modification destroys the purpose or function of the invention disclosed in the reference, one of ordinary skill in the art would not have found a reason to make the claimed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

The Office Action of October 31, 2007 cites column 13, line 66 – column 14 line 1 of Kuznetsov to teach that any number of translators can be implemented simultaneously, such that an entire set (or selected subset) of packets can be translated during runtime. The Office Action contends that this portion of Kuznetsov discloses *an output manager that facilitates at least one of* ***selectively pulling and pushing a subset of the transformed XML items***.... However, this contention is incorrect.

The cited portions of Kuznetsov simply teach any number of translators that can be implemented simultaneously to transform an entire set, or presumably a selected subset, of packets. However, no explanation is given as to how the selected subset can be generated; it is merely mentioned in passing. At the very least, Kuznetsov does not enable the subject matter recited by claim 1.

In addition, Kuznetsov does not enable translating a selected subset of packets in conjunction with ***a* transformer**. For instance, no explanation is given as to how "any number of translators" can facilitate at least one of selectively pulling and pushing a subset of the transformed XML items, provided by ***a transformer*** (singular). For example, presumably one or more of the "any number of translators" might be turned off to facilitate translating a selected subset of packets (such a result is speculation, however, it is not disclosed by Kuznetsov), but even that does not read on "***a translator*** ... and an output manager that facilitates at least one of selectively pulling and pushing a subset of the transformed XML items..." as turning off the translator of claim 1 would result in no items being transformed, arguably rendering the claimed subject matter inoperative. Therefore, as stated by the Court of Appeals for the Federal Circuit in *In re Gordon* (*supra*), one of ordinary skill in the art would not modify Kuznetsov to arrive at the claimed subject matter, because doing so would destroy the purpose of Kuznetsov. More specifically, if a single

transformer were utilized, as recited in independent claims 1, 19, and 38, turning off the transformer to generate a subset of transformed XML items would necessarily generate no XML items, destroying the purpose of translating XML items.

This argument was raised to Examiner Hillery in the telephonic interview of May 24, 2007. However, the argument was rejected because, in the examiner's view, independent claims 1 and 19 do not recite how a subset is selected, and that claiming the specifics of how the claimed subset is selected is required in order to overcome Kuznetsov based on applicants' representative's argument being true. Applicants' representative respectfully submits that this statement confuses the requirements for anticipation. Specifically, the requirement that each and every element of a claim be met by a reference and the requirement that the reference must teach the claimed subject matter are different requirements with different qualifications. A prior art reference must enable subject matter that it is cited as teaching, regardless of whether or not the words are parroted in one form or another.

> A prior art reference must be enabling as required for U.S. patents under 35 U.S.C. §112, first paragraph. *Paperless Accounting, Inc. v. Bay Area Rapid Transit Sys.*, 804 F.2d 659, 665, 231 USPQ 649, 653 (Fed. Cir. 1986), *cert. denied*, 480 U.S. 933 (1987). The description must enable a person with ordinary skill in the art not only to comprehend the invention but also to make it. *Paperless Accounting, Inc., v. Bay Area Rapid Transit Sys.*, 804 F.2d at 665, 231 USPQ at 653.

While it is true that an issued U.S. patent, such as Kuznetsov, used as a prior art reference is presumed to be valid, and that claims are presumed supported by an enabling disclosure (*see Ex parte* Goldgaber, 41 USPQ2d 1172, 1175 (B.P.A.I. 1996) (citing *in re* Lamberti, 545 F.2d 747, 751 n.2, 192 USPQ 278, 281 n.2 (C.C.P.A. 1976), that presumption extends ***only*** to the claimed subject matter of the U.S. patent. ***Kuznetsov does not claim*** the relevant aspects of independent claims 1 and 19, namely ***selectively pulling and pushing a subset of the one or more input XML items****, the subset of the one or more XML items is less than the whole one or more input XML items.* Therefore, applicants' representative submits that ***the presumption of enabling disclosure does not extend*** to "any

number of translators can be implemented simultaneously, such that an entire set (or selected subset) of packets can be translated during runtime" (Kuznetsov, column 13 line 66 through column 14 line 1) as cited by the Office Action of to render claims 1 and 19 obvious (Office Action dated October 31, 2007, p. 3, 4th paragraph). As stated above, the concept of transforming a subset of XML items is only mentioned in passing by Kuznetsov, and nowhere recited in any of its claims. Therefore, the cited portion of Kuznetsov is not enabled as applied to above-cited portions of claims 1 and 19, and thus does not teach or suggest the claimed subject matter and therefore cannot render the claimed subject matter obvious.

In view of the foregoing, it is believed that Kuznetsov does not teach or suggest each and every aspect of independent claims 1 and 19 (and claims 3, 4, and 5 that depend there from). Therefore, it is respectfully requested that this rejection be withdrawn.

III. Rejection of Claim 2 Under 35 U.S.C. §103(a)

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Kuznetsov (US 6,772,413 B2) as applied to claim 1 above and further in view of Omoigui (US 2003/0126136 A1). Withdrawal of this rejection is requested for at least the following reason: Kuznetsov and Omoigui, either alone or in combination, do not teach or suggest each and every aspect set forth in the subject claims.

> "Under 35 U.S.C. 103 where the examiner has relied on the teachings of several references, the test is whether or not the references viewed individually and collectively would have suggested the claimed invention to the person possessing ordinary skill in the art. It is to be noted, however, that citing references which merely indicated that isolated elements and/or features recited in the claims are known is not a sufficient basis for concluding that the combination of claimed elements would have been obvious. That is to say, there should be something in the prior art or a convincing line of reasoning in the answer suggesting the desirability of combining the references in such a manner as to arrive at the claimed invention... [I]t would not have been obvious to modify [the prior art] ... without using [the patent

> application's] claims as a guide. It is to be noted that simplicity and hindsight are not proper criteria for resolving the issue of obviousness." *Ex parte Hiyamizu*, 10 USPQ2d 1393 (BPAI 1988).

Claim 2 recites: the system of claim 1, the transformer comprises an action frame stack that holds one or more actions, an event state machine that tracks state associated with transforming the one or more XML items and an event processor that receives events generated in processing the one or more actions stored in the action frame stack. The Office Action contends that Omoigui teaches "the system provides support for authentication, authorization, auditing, data privacy, data integrity, availability, and non-repudiation by employing standards such as WS-Security. WS-Security provides a platform for security with XML Web Service applications using standards in the XML Web Serve protocol stack. This includes method calls from clients, support for digital signatures, authenticating the calling user before granting access to an Agency's Semantic Network and XML Web Service methods, etc." (Omoigui, paragraph 0367).

The Office Action further contends that the cited portion of Omoigui reads on the limitation recited in claim 2 (Office Action dated October 31, 2007, p. 7, 4th paragraph). This is incorrect. An "XML Web Service protocol stack" is not equivalent to an action frame stack that holds one or more actions. Applicants' representative submits that an action stack, as recited in claim 2, can be considered analogous (although not equivalent) to a queue that can store actions while other actions are being processed. It is unclear from Omoigui or from the Office Action what the XML Web Service protocol stack of Omoigui is. However, it is submitted that a Web services protocol stack is merely a collection of protocols for to define standards of electronic communication. However, below is cited at least one article to support this notion. According to Wikipedia, a Web services protocol stack is:

> "a collection of computer networking protocols that are used to define, locate, implement and make Web services interact with each other. The Web service protocol stack mainly comprises four areas: a (Service) Transport Protocol responsible for transporting messages between network applications and includes protocols such as HTTP, SMTP, FTP ... an (XML) Messaging Protocol response for encoding messages in a common XML format so that they can be understood at either end of a network connection ... a (Service) Description Protocol used for describing the public interface to a

> specific web service ... and a (Service) Discovery Protocol that centralizes services into a common registry such that network web services can publish their location and description, and makes it easy to discover what services are available on the network..."

Accordingly, applicants' representative submits that the XML Web Service protocol stack of Omoigui is merely a collection of computer networking protocols pertinent to providing "method calls from client [device]s, support for digital signatures, authenticating the calling user before granting access to an Agency's Semantic Network and XML Web Service methods" (Omoigui, paragraph 0367). It is manifest, however, that a set of protocols do not equate to, teach, or suggest *an action frame stack that holds one or more actions*, as recited by claim 2. Protocols define syntax, organization, structure and the like for software code. They do not, in and of themselves, perform activities such as holding one or more actions, as does the action frame stack of claim 2.

In addition to the foregoing, Omoigui does not include *an event state machine that tracks state associated with transforming the one or more XML items* and does not include *an event processor that receives events generated in processing the one or more actions stored in the action frame stack.* Neither does Omoigui state that these items are included within the XML Web Service protocol stack. Such contention appears for the first time in the Office Action, unsupported by the cited references. Omoigui appears to recite an XML Web Service security provision that can encrypt method calls, support digital signatures, authenticate calls, and the like, according to standards provided by the XML Web Services protocol stack. This, however, does not remotely meet the limitations of claim 2.

In view of the foregoing, it is believed that neither Kuznetsov nor Omoigui, alone, in combination, teach, or suggest each and every aspect of claim 2. Therefore, it is respectfully requested that this rejection be withdrawn.

IV. Rejection of Claims 6-18 Under 35 U.S.C. §103(a)

Claims 6-18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kuznetsov (US 6,772,413 B2) as applied to claim 1 above and further in view of ADO.NET (English Translation). Withdrawal of this rejection is requested for at least the following reasons: ADO.NET is not prior art with respect to the subject application, and ADO.NET and Kuznetsov,

either alone or in combination, do not teach or suggest each and every aspect set forth in the subject claims.

> Prior art disclosures on the Internet or on an on-line database are considered to be publicly available as of the date the item was publicly posted. Absent evidence of the date that the disclosure was publicly posted, *if the publication itself does not include* ***a publication date (or retrieval date), it cannot be relied upon as prior art*** under 35 U.S.C. 102(a) or (b). MPEP §2128.

ADO.NET appears to be an Internet document, presumably retrieved from an online database, cited by the Office Action dated October 31, 2007 to render obvious portions of claims 6-18. However, ADO.NET provides no publication date. Further, the Office Action provides no date as to when the reference was retrieved. It is submitted, therefore, that ADO.NET cannot be relied upon as prior art under 35 U.S.C. 102(a) or (b), or by extension 35 U.S.C. 103(a).

The office action appears to rely on a date printed on the cover sheet of ADO.NET, namely, March 6 – 8, 2001. However, this date is insufficient proof that the information was publicly available on the Internet as of March 6 – 8, 2001. It is submitted that any desired date can be printed on an internet document; the printing of such date, however, does not imply that the document was available to the public (*e.g.*, hosted by an Internet site and properly catalogued by a search engine) on that date, as required by the MPEP. As a hypothetical example, a date of March 6-8, 2015 can easily be printed on any internet document, just as March 6-8, 2001 appears on the cover of ADO.NET. However, clearly March 6-8, 2015 could not correspond to a date of public availability for the hypothetical example, because such point in time has not yet come to pass. It is submitted that any arbitrary date on the cover of an Internet reference, without more information, is insufficient proof of a date of public availability.

Further to the above, the cited date (March 6-8, 2001) appears on the cover of the reference in conjunction with a name of a city in Denmark (Copenhagen), under a title ADO.NET. It is submitted, therefore, that this date is merely a date on which a conference on the ADO.NET subject matter was held at Copenhagen, Denmark, and further that appearance of such date on the reference cover is insufficient proof of a publication date, a date of public availability, or a retrieval date of the ADO.NET reference. It is submitted, therefore, that the only date that can be relied on for the ADO.NET reference is a date that the examiner personally downloaded it from the Internet, or the Office obtained it. Further to this end, it is noted that another date does

appear on the cover of ADO.NET, specifically March 2005, in conjunction with UNITED STATES PATENT AND TRADEMARK OFFICE Washington, D.C. It is further submitted that ADO.NET was not obtained by the Patent Office until March 2005, and therefore this reference should be accorded a date no earlier than March 2005, unless the examiner can prove that ADO.NET was publicly available prior to the filing date of the subject application, as required by the above-cited portion of the MPEP.

In addition to the foregoing, the Office Action dated October 31, 2007 fails to establish a *prima facie* case of obviousness because neither Kuznetsov nor ADO.NET teach or suggest all the elements of claim 1, from which claims 6-18 depend, or the additional elements recited by claims 6, 8, 9, 10, 12, 15-18, and newly added claim 38. As stated above, Kuznetsov does not recite the elements of independent claim 1, specifically, ***a transformer*** *that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats; and an output manager that facilitates at least one of* ***selectively pulling and pushing a subset of the transformed XML items***. As discussed above, he Office Action recites column 13 lines 66 through column 14 line 1 of Kuznetsov to render obvious "an output manager that facilitates at least one of selectively pulling and pushing a subset of the transformed XML items".

The cited portions of Kuznetsov simply teach any number of translators that can be implemented simultaneously to transform an entire set of packets; no explanation is given as to how a selected subset of packets can be translated. As discussed above, Kuznetsov does not enable the subject matter recited by claim 1. Therefore, "any number of … translators can be implemented simultaneously, such that an entire set (or selected subset) of packets can be translated" (Kuznetsov, column 13 line 66 through column 14 line 1) does not teach or suggest the subject matter of claim 1. In addition, ADO.NET is silent with respect to this element of claims 6-18, and therefore does not cure the deficiencies of Kuznetsov.

Furthermore, with regard to claim 6, the Office Action cites ADO.NET to teach "an XpathNavigator is created to abstract data from the xml data set via an XpathNodeIterator by employing a loop", and contends that this portion of ADO.NET renders obvious: *an input abstractor that exposes data stored in the one or more data stores in a common representation* (Office Action dated October 31, 2007, p. 8, 5th paragraph). This would require that all abstraction of data employing a loop exposes the abstracted data in a common representation.

Nothing in the reference supports this result, and the Office Action does not suggest it either. Therefore, the cited portion of ADO.NET is simply insufficient to meet the subject matter recited in claim 6.

With regard to claim 8, the Office Action recites page 19 of ADO.NET to teach an XpathNavigator is created to abstract data from the xml data set (Office Action dated October 31, 2007, p. 9, 5th paragraph). The Office Action claims that this subject matter renders obvious "the input abstractor exposes the data stored in the one or more data stores as a data model and infoset". This statement is logically incorrect, however, as it would require all data abstracted from an xml data set to be exposed in one or more data stores as a data model and infoset. This is not the case. Consequently, the cited portion of ADO.NET is insufficient to meet the claimed subject matter recited by claim 8.

With regard to claim 9, the Office Action recites page 19 of ADO.NET to teach an XpathNavigator is created to abstract data from the xml data set and sends the data to an XSLT (Office Action dated October 31, 2007, p. 10, 2nd paragraph). The Office Action claims that this subject matter teaches "the input abstractor provides a cursor model over data stored in a data store to facilitate presenting a stream of nodes to the transformer", as recited by claim 9. It is not apparent how abstracting data and sending said data to an XSLT meets "providing a cursor model over data stored in a data store to facilitate presenting a stream of nodes to a transformer". This would require all abstraction of data to necessarily include "providing a cursor model over data stored in a data store to facilitate presenting a stream of nodes". The ADO.NET reference certainly does not disclose this proposition, and no other reference cited in the Office Action suggests this result either. Consequently, ADO.NET is insufficient to meet the subject matter of claim 9.

With regard to claim 10, the Office Action recites page 19 of ADO.NET to teach an XpathNavigator is created to abstract data from the xml data set (Office Action dated October 31, 2007, p. 10, 5th paragraph). The Office Action contends that this cited portion of ADO.NET discloses "the input abstractor provides a virtual node that can be employed to traverse the stream of nodes" recited by claim 10. However, this result also requires all data abstracted from an xml data set to "provide a virtual node that can be employed to traverse the stream of nodes." ADO.NET does not disclose this proposition, and no other reference cited in the Office Action

suggests this result either. Consequently, the cited portion of ADO.NET is insufficient to meet the subject matter of claim 9.

With regard to claim 12, the Office Action recites page 19 of ADO.NET to teach "SQL is used to query xml items and store them to an XML data set and that each node in the xml dataset is visited by employing an XpathNodeIterator (Office Action dated October 31, 2007, p. 11, 5th paragraph). The Office Action contends that this cited portion of ADO.NET discloses "a node selection abstractor that dynamically constructs a subset of input XML items from a set of input XML items, the subset of input XML items are responsive to a query". An SQL query does not necessarily dynamically construct a subset of input XML items from a set of input XML items. Such a result would be necessary or at least implicit for the cited portion of ADO.NET to teach or suggest the subject matter of claim 12.

In addition, nothing in ADO.NET or in Kuznetsov provides motivation to combine an SQL query with constructing a subset of input XML items from a set of input XML items in conjunction with a transformer that transforms one or more input XML items in a first format to one or more transformed XML items in one or more second XML formats, as recited by claim 12.

> "Under 35 U.S.C. 103 where the examiner has relied on the teachings of several references, the test is whether or not the references viewed individually and collectively would have suggested the claimed invention to the person possessing ordinary skill in the art. It is to be noted, however, that citing references which merely indicated that isolated elements and/or features recited in the claims are known is not a sufficient basis for concluding that the combination of claimed elements would have been obvious. That is to say, there should be something in the prior art or a convincing line of reasoning in the answer suggesting the desirability of combining the references in such a manner as to arrive at the claimed invention... [I]t would not have been obvious to modify [the prior art] ... without using [the patent application's] claims as a guide. It is to be noted that simplicity and hindsight are not proper criteria for resolving the issue of obviousness." *Ex parte Hiyamizu*, 10 USPQ2d 1393 (BPAI 1988).

The Office Action contends that doing so would have been obvious because it provides a benefit of "explicit implementation of XPath via source code." (Office Action dated October 31, 2007,

p. 11, 6th paragraph and p. 12 1st paragraph.) However, constructing a benefit in hindsight does not meet the requirement of "something in the prior art or a convincing line of reasoning in the answer suggesting the desirability of combining the references in such a manner as to arrive at the claimed invention…." No reference cited by the Office Action acknowledges such a benefit, nor does the Office Action provide any credence to the fact that such a benefit would have been considered by one of skill in the art prior to the filing date of the subject application. Consequently, the cited references neither teach nor suggest all subject matter recited by claim 12.

With regard to claim 15, the Office Action recites pages 18-19 of ADO.NET to teach "that SQL is used to query xml items and store them to an XML data set" (Office Action dated October 31, 2007, p. 13, 2nd paragraph). The Office Action further contends that this subject matter discloses "an optimized data store that stores one or more XML items in a manner that facilitates minimizing processing associated with constructing the subset of input XML items via a query." However, this contention requires that all SQL queries of xml items stored to an XML data set facilitates minimizing processing associated with constructing the subset of input XML items via a query. Nothing in ADO.NET proposes this requirement, and the Office Action gives no addition grounds to do so either. In addition, as stated above with regard to claim 12, the Office Action provides insufficient motivation to combine Kuznetsov and ADO.NET, as nothing in either reference and nothing else provided in the Office Action suggests that one of skill in the art would be motivated to combine the references by "the benefit of explicit implementation of XPath via source code". This conclusion seems to be manufactured wholly at first instance by the Office Action; the cited art does not support it. Consequently, it is submitted that the cited references neither teach nor suggest all subject matter recited by claim 12, nor does there exist a sufficient motivation to combine such references as required by MPEP §706.02.

With regard to claims 16 and 18 the Office Action recites pages 18-19 of ADO.NET to teach that an "XPath document is created and used to store and manipulate the xml data set" (Office Action dated October 31, 2007, p. 13, 5th paragraph and p. 14, 6th paragraph). The Office Action contends that this cited portion discloses "the optimized data store stores data in a data representation format that facilitates optimizing an XPath query". Similar to analogous conclusions cited with respect to other claims, stated above, this contention is logically flawed in that it requires that every creation of an XPath document that stores and manipulates an xml data

set to necessarily include a data representation format that facilitates optimizing an XPath query. This result would logically negate any optimization of the XPath query, because if all queries involved the same data representation that optimizes the query, none would be optimal over another. Consequently, ADO.NET is insufficient to teach the subject matter recited by claim 16.

In addition to the foregoing, newly added independent claim 38 recites ***a transformer*** *that receives a plurality of input XML items and transforms the input XML items from a first format to one or more second XML formats*; ***an output manager*** *that* ***facilitates at least one of selectively pulling and pushing a subset of the transformed XML items, the subset having fewer XML items than the plurality of input XML items, to provide a selective streaming output****, the selective streaming output is provided to a memory*; and ***a node selection abstractor*** *that* ***dynamically constructs the plurality of input XML items*** *from a set of input XML items, the plurality of input XML items are* ***responsive to a query***. It is submitted that neither Kuznetsov nor ADO.NET teach or suggest the above-cited aspects of claim 38. Specifically, neither Kuznetsov nor ADO.NET teach or suggest a transformer in conjunction with an output manager that facilitates at least one of selectively pulling and pushing a subset of transformed XML items. Further, neither Kuznetsov nor ADO.NET teach or suggest a node selection abstractor that dynamically constructs the plurality of input XML items … responsive to a query. Accordingly, it is submitted that claim 38 recites patentable subject matter in view of the submitted references.

In view of the foregoing, it is believed that neither Kuznetsov nor ADO.NET, alone or in combination, teach or suggest each and every aspect of independent claims 1 and 19 (and claims 2-18 that depend there from, or by extension claim 38). Therefore, it is respectfully requested that this rejection be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited. In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP296US]. Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 48,064

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731